UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/606,024	06/25/2003	R. Victor Klassen	D/A2218I1	3942

62095 7590 04/28/2010
FAY SHARPE / XEROX - ROCHESTER
1228 EUCLID AVENUE, 5TH FLOOR
THE HALLE BUILDING
CLEVELAND, OH 44115

EXAMINER	
ROBINSON, MYLES D	
ART UNIT	**PAPER NUMBER**
2625	
MAIL DATE	**DELIVERY MODE**
04/28/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/606,024	KLASSEN ET AL.	
	Examiner	Art Unit	
	Myles D. Robinson	2625	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *12 February 2010*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1 - 20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1 - 20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *16 July 2009* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

Response to Amendment

1. Applicant's amendment was received on 2/12/2010, and has been entered and made of record. Currently, **claims 1 – 20** are pending.

2. Applicant's request for reconsideration of the finality of the rejection of the last Office action is persuasive and, therefore, the finality of that action is withdrawn.

Response to Arguments

3. Applicant's arguments (*see Remarks 2/12/2010 [pages 8 – 10]*) with respect to the rejection of **claims 1 – 19** under 35 U.S.C. §103(a) and the rejection of **claim 20** under 35 U.S.C. §102(e) have been fully considered and are persuasive. Therefore, the rejection has been withdrawn. However, upon further consideration, a new ground(s) of rejection is made in view of **Cyr *et al.*** (U.S. Patent No. 5,819,014) and in view of **IBM Technical Disclosure** NN9308637.

Regarding **claims 1 and 20**, the Applicant argues that **Christiansen *et al.*** (U.S. Patent Application Publication No. 2004/0114170) in view of **Yomogizawa** (Japanese Patent No. 09-050354) does not disclose, teach or suggest a virtual disk remote transfer system that comprises an intermediary storage for data transfer that is implemented by providing a shared memory interface (*see Remarks 2/12/2010 [pages 9 – 10]*).

However, Cyr in view of the IBM Technical Disclosure does disclose a virtual disk remote transfer system that comprises an intermediary storage for data transfer that is implemented by providing a shared memory interface.

Therefore, the Applicant's arguments regarding claims 1 and 20 are considered not persuasive. Please cite rationale of the grounds of rejection below for further explanation.

Claim Rejections - 35 USC § 112

4. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

5. ***Claims 1 – 19*** are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claims 1 and 16 recite the limitation "the transfer system data" in lines 10 and 9, respectively. There is insufficient antecedent basis for this limitation in the claim. All claims dependent upon these claims suffer the same deficiency and, therefore, are rejected as well.

6. A broad range or limitation together with a narrow range or limitation that falls within the broad range or limitation (in the same claim) is considered indefinite, since the resulting claim does not clearly set forth the metes and bounds of the patent protection desired. See MPEP § 2173.05(c). Note the explanation given by the Board of Patent Appeals and Interferences in *Ex parte Wu*, 10 USPQ2d 2031, 2033 (Bd. Pat. App. & Inter. 1989), as to where broad language is followed by "such as" and then narrow language. The Board stated that this can render a claim indefinite by raising a

question or doubt as to whether the feature introduced by such language is (a) merely exemplary of the remainder of the claim, and therefore not required, or (b) a required feature of the claims. Note also, for example, the decisions of *Ex parte Steigewald*, 131 USPQ 74 (Bd. App. 1961); *Ex parte Hall*, 83 USPQ 38 (Bd. App. 1948); and *Ex parte Hasche*, 86 USPQ 481 (Bd. App. 1949). In the present instance, **claims 1 and 16** recite the broad recitation "a virtual disk transfer system", and the claim also recites "the virtual disk *remote* transfer system" which is the narrower statement of the range/limitation. All claims dependent upon these claims suffer the same deficiency and, therefore, are rejected as well.

Claim Rejections - 35 USC § 103

7. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

8. ***Claims 1, 3, 5, 6, 11, 13 – 17 and 20*** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Cyr *et al.*** (U.S. Patent No. 5,819,014) and in view of **Christiansen *et al.*** (U.S. Patent Application Publication No. 2004/0114170) and further in view of **IBM Technical Disclosure** NN9308637.

Referring to **claim 1**, Cyr discloses a method of operating a printing system for parallel processing a print job with a plurality of processing nodes into a printer ready format for printing the print job (*see Figs. 2 – 4 wherein the printer architecture utilizes network resources to distribute printer controller and translator functions and thereby process several print jobs [e.g. pre-rasterized images, which are equivalent to printer*

ready format for printing the print job] in parallel and wherein translators 12, 14, 16 are equivalent to a plurality of processing nodes [Abstract, column 3, lines 24 – 35 and 59 – 67]), said processing nodes communicating with a virtual disk transfer system (*see Figs. 3, 4, 6 and 10 wherein user interface job controller 66 utilizes VMS [virtual machine system] operating system and sends commands to print controller 25 as well as control process 67 of translator subsystems 12, 14, 16 to ensure an orderly flow of information between the components [column 4, lines 25 – 32, column 6, lines 19 – 24 and column 10, lines 57 – 67]*), comprising:

selectively storing the jobs and print-ready pages in the virtual disk remote transfer system (*see Figs. 4, 6 and 8 and 10 wherein print jobs or pre-rasterized images can be processed and stored while awaiting printing in storage device 11 in state 94 [column 5, lines 1 – 23, column 6, lines 19 – 31, 49 – 55 and column 7, lines 56 – 63] and see Figs. 4, 6 and 7wherein if the source document is pre-rasterized image in storage device 11, the print controller enters state 73 to access the stored image for transmission to printer interface 18 of state 76 [column 7, lines 18 – 22] such that the VMS operating system of job controller 66 controls storing/printing of jobs and print-ready pages [column 6, lines 19 – 24 and column 10, lines 57 – 67]*),

assigning the job chunks to respective processing nodes for parallel processing the jobs into the printer-ready format (*see Figs. 2, 4 and 6 wherein printer controller 25 assigns one of translators 12, 14, 16 to printing of a print request [column 6, lines 31 – 38 and 42 – 45]*), and

printing the print job (*see Figs. 2 and 4, printer 20 [column 3, lines 29 – 24 and column 5, lines 11 – 13], see Fig. 5, printer 48 [column 5, line 66 – column 6, line 4], see Fig. 6, printer engine 29 and see Fig. 7, state 76 [column 7, lines 23 – 31]*) but does not explicitly disclose the method further comprising splitting the print job into a plurality of job chunks, wherein the chunks range in size-from at least one page to the entire size of the print job, wherein pages comprising the chunks are selected in accordance with predetermined selection factors for optimizing page printing processing efficiency, wherein the transfer system data comprises an intermediary storage for data transfer to selected processing nodes including a RAM and a physical disk, and where the virtual disk system is implemented by providing a shared memory interface, assigning the job chunks to respective processing nodes for parallel processing the job chunks into the printer-ready format, monitoring available space in the virtual disk transfer system including detecting a data overflow in the RAM and storing new data in the physical disk until data storage in the RAM is available.

Christiansen discloses a method of operating a printing system for parallel processing a print job (*see Fig. 1 wherein raster image processing (RIP) system 100 processes print job 103 in parallel [paragraphs 0020 – 0021 and 0035]*) with a plurality of processing nodes into a printer ready format for printing the print job (*see Figs. 1 and 2 wherein RIP engines 109 rip print job 103 into a printer ready format for RIPped output file 123 for output to print device 126 [paragraphs 0021 – 0022 and 0025 – 0026]*), comprising:

splitting the print job into a plurality of job chunks (*see Fig. 2, partition manager 176 and see Fig. 14, steps 519 – 523 [paragraphs 0064 – 0065]*), wherein the chunks range in size from at least one page to the entire size of the print job (*paragraphs 0021 and 0023 wherein the partitions of print job 103, which are analogous to the plurality of job chunks of the print job, are defined as a subset of the total number of pages in a print job that may be consecutive or nonconsecutive*), wherein pages comprising the chunks are selected in accordance with predetermined selection factors for optimizing page printing processing efficiency (*see Fig. 9 wherein a user provides pipeline acceptance criteria 299 for a respective pipeline 113 such that the criteria 299 is analogous to predetermined factors [paragraphs 0041 – 0042], see Figs. 15A – 15B wherein the system determines partitions of the print job based upon criteria 299 [i.e. predetermined factors] [paragraphs 0067 – 0071] and see Figs. 1 and 16 wherein the RIPped output file 123 is then applied to a print device 126 for printing in box 716 [paragraphs 0026 and 0083]*),

assigning the job chunks to respective processing nodes for processing the job chunks into the printer-ready format (*see Figs. 2 and 14 wherein print job preprocessor 173 determines which pipeline 133, which comprises RIP engines 109, to which the print job 103 is to be applied in step 519 and then transfers the print job to partition manager 176 [paragraph 0064] and see Figs. 2 and 15B wherein then partition manager 176 applies partition to the pipeline 113 for RIPping in step 633 [paragraph 0073]*), and

printing the print job (*see Figs. 1 and 16 wherein the RIPped output file 123 is then applied to a print device 126 for printing in box 716 [paragraphs 0026 and 0083]*)

but does not explicitly disclose the method further wherein the transfer system data comprises an intermediary storage for data transfer to selected processing nodes including a RAM and a physical disk, and where the virtual disk remote transfer system implemented by providing a shared memory interface, and the method further comprising monitoring available space in the virtual disk transfer system including detecting a data overflow in the RAM and storing new data in the physical disk until data storage in the RAM is available.

IBM Technical Disclosure discloses the method wherein the virtual transfer system (*see Title*) data comprises an intermediary storage for data transfer to selected processing nodes including a RAM and a physical disk (*"Separate disclosures have shown [that] a shared RAM disk can be used to provide fast access to non-permanent 'disk' storage [e.g. physical disk]"*), and where the virtual disk remote transfer system implemented by providing a shared memory interface (*"Disclosed is a program to create a shared disk for a cluster of processors which have separate I/O subsystems but a global shared memory. The program uses the shared memory to pass requests and data for I/O between processors, making it look like there is one shared disk rather than several disks that can only be accessed by the local processor." and "Separate disclosures have shown... that a shared virtual disk can be created to support shared access by all processors to disks, both local and remote, in the cluster"*), and the method further comprising:

monitoring available space in the virtual disk transfer system including detecting a data overflow in the RAM (*"A WRITE operation searches the shared memory cache*

[e.g. RAM] for the block that is being written. If it is not found in the cache, an empty cache block is selected and the data is copied into that cache block" wherein the conditional "if/if not" statement regarding the state of the cache is equivalent to a detection of available virtual space in RAM) and storing new data in the physical disk until data storage in the RAM is available (*"The write can then be made to by the appropriate processor (write-through) or it can be delayed until later, depending upon the need of the system for integrity of the disk contents in the event of a system failure" and "However, the system administrators can have direct control over the write-through (or not) policy giving them direct control over the choice between speed (no write-through) and data integrity (write-through)" wherein the "write-through" during the write operation is equivalent to storing new data in the physical data*).

Cyr and Christiansen are combinable because they are from the same field of endeavor, being high-speed image processing systems capable of printing large print jobs using multiple RIP engines. At the time of the invention, it would have been obvious to one of ordinary skill in the art to include splitting the job into a plurality of job chunks for processing of the print job at multiple RIP engines. The suggestion/motivation for doing so would have been to save time by partitioning very large documents, as suggested by Christiansen (*see Abstract and paragraph 0001*). See MPEP 2143 G.

Cyr and the IBM Technical Disclosure are combinable because they are from the same field of endeavor, being virtual disk remote transfer systems. At the time of the invention, it would have been obvious to one of ordinary skill in the art to include a write-

through operation for storing new data in the physical disk. The suggestion/motivation for doing so would have been to ensure data integrity at the administrator's discretion, as suggested by IBM Technical Disclosure (*"The write can then be made to by the appropriate processor (write-through) or it can be delayed until later, depending upon the need of the system for integrity of the disk contents in the event of a system failure"* and *"However, the system administrators can have direct control over the write-through (or not) policy giving them direct control over the choice between speed (no write-through) and data integrity (write-through)"*). See MPEP 2143 G.

Furthermore, both Cyr and the IBM Technical Disclosure teach improvements in the same way as the claimed invention, wherein the virtual disk remote transfer system is ideally designed for parallel systems (*see Cyr [Title, Abstract, column, 2, lines 19 – 31, column 3, lines 59 – 67 and column 4, lines 9 – 13] and see IBM ["To improve the ability of the processors to work together to solve common problems, it is useful to provide I/O devices that can be accessed by all processors equally. Such a shared I/O device would allow processes which are working on a common problem to use the shared I/O device for storage of common data, while still allowing these processes to be run on any of the available processors."*). One of ordinary skill in the art could have applied the known improvement techniques in the same way to the other technique and the results would have been predictable to one of ordinary skill in the art. See MPEP 2143 C.

Referring to **claim 3**, Christiansen discloses the method further wherein the splitting step is preformed by a splitter (*see Figs. 2 and 15A – 15B, partition manager*

176 [paragraphs 0064 – 0065]) and further comprising the step of withholding chunk destinations from the splitter (*see Figs. 2 and 16 wherein RIP handler 179 receives the partitioned print job [i.e. plurality of job chunks] and assigns those partitions to RIP engines 209 for processing in such a manner that RIP handler 179, which determines the chunk destinations [i.e. pipeline 113 comprising RIP engines 109], functions properly without revealing to partition manager 176, which splits the print job into chunks, which one(s) of RIP engines 109 the partitions will be assigned [i.e. withholding chunk destinations from the splitter] [paragraphs 0074 – 0083]*).

Referring to **claim 5**, Christiansen discloses the method further wherein the predetermined factors used to determine the size of <u>the print</u> job further includes the total number of pages within the print job (*see Figs. 1 and 14 wherein RIP manager 106 first determines a page count of the total number of pages of print job 103 and then determines a number of partitions [i.e. chunks] for the print job 103 assuming the page count can be determined in steps 509, 516, 519 and 523 [paragraph 0021]*).

Referring to **claim 6**, Christiansen discloses the method further wherein the predetermined factors used to determine the size of <u>the print</u> job further includes the total number of bits within the print job (*see Fig. 9 wherein a user provides pipeline acceptance criteria 299 for a respective pipeline 113 such that the criteria 299 is analogous to predetermined factors [paragraphs 0041 – 0042] and wherein one acceptance criteria 299 includes "all jobs with a file size between" such that file size is the equivalent to the total number of bits within the print job [see Fig. 9] and see Figs. 15A – 15B wherein the system determines partitions of the print job based upon criteria*

299 [i.e. predetermined factors] [paragraphs 0067 – 0071] and see Figs. 1 and 16 wherein the RIPped output file 123 is then applied to a print device 126 for printing in box 716 [paragraphs 0026 and 0083]).

Furthermore, Christiansen discloses the method further wherein the predetermined factors used to determine the size of the print job further includes the total number of pages within the print job (*see Figs. 1 and 14 wherein RIP manager 106 first determines a page count of the total number of pages of print job 103 and then determines a number of partitions [i.e. chunks] for the print job 103 assuming the page count can be determined in steps 509, 516, 519 and 523 [paragraph 0021]*). One of ordinary skill in art would recognize that the pages of the print job are inherently stored as bits of data. In other words, the total number of pages within the print job is inherently represented as a certain number of bits for that print job.

Referring to **claim 11**, Christiansen discloses the method further wherein separate print queues are used (*see Figs. 1, 2 and 15B wherein partition manager 176 applies the print job 103 to a pipeline 113 such that the print job is placed into a queue associated with the respective pipeline 113, etc. or an alternative general queue for all print jobs in step 633 [paragraph 0073]*). Claim scope is not limited by claim language that suggests or makes optional but does not require steps to be performed, or by claim language that does not limit a claim to a particular structure. Specifically, the claim language "may be defined" suggests optional functionality or utility. See MPEP §2111.04, §2143 G, §2144.04 I.

Referring to **claim 13**, Christiansen discloses the method further wherein the parallel processing system incorporates load balancing to spread the workload out evenly among <u>associated</u> print devices (*see Fig. 1 wherein RIP manager 106 assigns partitions to one or more RIP engines 109 for raster image processing [paragraphs 0022 and 0025]*).

Referring to **claim 14**, Christiansen discloses the method further wherein the parallel processing system incorporates auto discovery to evaluate the availability of hardware resources (*see Fig. 15A, step 556 [paragraph 0066]*).

Referring to **claim 15**, Christiansen discloses the method further wherein the splitting functionality may have user selected status associated with each partition (*see Fig. 2 wherein RIP manager 106 tracks the status [e.g. unassigned, assigned, completed, or their equivalents] of the RIPping of each of the partitions 189 [paragraphs 0021, 0022, 0024 and 0031] and see Figs. 5 – 6 wherein user interface 196a allows the user to manipulate these statuses by reassigning particular partitions to another pipeline 113 [paragraphs 0036 – 0038]*).

However, Christiansen does not explicitly disclose the method further wherein the statuses have the specific names maximum, recommended and allocated.

The Examiner asserts that these are mere design changes. At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to modify the names of the statuses associated with each partition during parallel processing. Applicant has not disclosed how these particular status names provide an advantage, is used for a particular purpose, or solves a stated problem. One of ordinary

skill in the art, furthermore, would have expected Applicant's invention to perform equally well with using the status names taught in Christiansen or their equivalents because the status names “maximum,” "recommended” and “allocated” could be applied to some other aspect of splitting functionality under the broadest reasonable interpretation.

For example, the status “allocated" could be equivalent to the status “assigned” which represents that the allocated partition has already been assigned for ripping, the status “recommended” could be equivalent to the status "unassigned” which represents that this unassigned partition is one recommended next for ripping, and the status “maximum" could be equivalent to the status “completed” which represents that the maximum amount of the partition has been completely rendered. Claim scope is not limited by claim language that suggests or makes optional but does not require steps to be performed, or by claim language that does not limit a claim to a particular structure. Specifically, the claim language "may have" suggests optional functionality or utility. See MPEP §2111.04, §2143 G, §2144.04 I.

Therefore, it would have been obvious to one of ordinary skill in this art to modify the names of the statuses taught by Christiansen to obtain the invention as specified in claim 15.

Referring to **claim 16**, the rationale provided above in the rejection of claim 1 is incorporated herein. In addition, the method of claim 1 includes the elements and limitations of the method of claim 16. Furthermore, Christiansen discloses a plurality of parallel processors (*see Fig. 1 wherein raster image processing (RIP) system 100*

processes print job 103 in parallel [paragraphs 0020 – 0021 and 0035] and see Figs. 1 and 2 wherein RIP engines 109 rip print job 103 for output to print device 126 [paragraphs 0021 – 0022 and 0025 – 0026]).

Referring to **claim 17**, Christiansen discloses the method further wherein the parallel processing comprises at least three processors (*see Figs. 1 and 3 wherein RIP manager 106 assigns partitions to one or more RIP engines 109 [paragraph 0022]*) connected to at least three separate memories (*see Fig. 3 wherein each RIP engine has its own memory 206 [paragraph 0032]*).

Referring to **claim 20**, Christiansen discloses a method of operating a printing system for parallel processing job (see Fig. 1 wherein raster image processing (RIP) system 100 processes print job 103 in parallel [paragraphs 0020 – 0021 and 0035]) comprising the steps of:

inputting a print job (*see Figs. 2 and 14 wherein print job preprocessor 173 awaits input of print job 103 in step 503 [paragraphs 0062 – 0063]*),

storing the print job in a spooling system (*see Figs. 2 and 14 wherein print job 103 is stored in a queue or other portion of memory 146 in step 506 [paragraphs 0028 – 0029, 0034 – 0035 and 0063]*),

determining the language, size and location of the print job (*see Figs. 1, 9 and 14 wherein print job preprocessor 173 determines which pipeline 113 to which the print job 103 is to be applied [i.e. location of the print job] based upon the priorities assigned to each of the existing pipelines 113 as well as the pipeline acceptance criteria 299 [paragraph 0064] and see Fig. 9 wherein pipeline acceptance criteria 299 for a*

respective pipeline 113 include specifying a specific page description language for the print job that will be accepted [i.e. language of print job] and specifying a range of file sizes for acceptable print jobs [paragraphs 0041 – 0042]),

advising the supervisor to select splitter (*see Fig. 9 wherein a user provides pipeline acceptance criteria 299 for a respective pipeline 113 such that the criteria 299 is analogous advisement to RIP manager 106 [i.e. supervisor] [paragraphs 0041 – 0042] and see Figs. 15A – 15B wherein the system determines partitions of the print job based upon criteria 299 [i.e. predetermined factors] [paragraphs 0067 – 0071]*),

splitter advised of job location and chunk parameters (*see Figs. 9 and 14 wherein print job preprocessor 173 uses pipeline acceptance criteria 299, which are equivalent to advisement of chunk parameters, to advise partition manger 176 [i.e. splitter] in determining which pipeline 113 to which the print job is to be applied [i.e. job location] in step 519 [paragraphs 0041 – 0042 and 0064]*),

assigning chunk to rasterized image processor (RIP) nodes (*see Figs. 2 and 14 wherein print job preprocessor 173 determines which pipeline 133, which comprises RIP engines 109, to which the print job 103 is to be applied in step 519 and then transfers the print job to partition manager 176 [paragraph 0064] and see Figs. 2 and 15B wherein then partition manager 176 applies partition to the pipeline 113 for RIPping in step 633 [paragraph 0073]*),

splitting the job into chunks (*see Fig. 2, partition manager 176 and see Fig. 14, steps 519 – 523 [paragraphs 0064 – 0065]*),

sending chunks to RIP node (*see Figs. 1, 2 and 15B wherein then partition manager 176 applies partition to the pipeline 113 for RIPping in step 633 [paragraphs 0021 and 0073]*),

maintain chunk order by collector (*see Figs. 1 and 16 wherein RIP manager 106 re-aggregates the RIPped partitions 119 into a single RIPped output file 123 in step 709 [paragraphs 0026 and 0083]*),

advising supervisor of chunk completion by splitter (*see Figs. 1, 2 and 16 wherein RIP manager 106 tracks a status 189 of the RIPping of each of the partitions based upon the partition specification in steps 673, 693, 706, 713 [paragraphs 0021, 0076 – 0077, 0079 and 0082 – 0083]*),

advising collector of page to chunk association (*see Figs. 1, 2 and 16 wherein RIP manager 106 comprising RIP handler 179 re-aggregates the RIPped partitions 119 into a single RIPped output file 123 in step 709 [paragraphs 0026 and 0083]*),

parallel processing of chunks by RIP nodes (*see Fig. 1 wherein raster image processing (RIP) system 100 processes print job 103 in parallel [paragraphs 0020 – 0021 and 0035] and see Figs. 1 and 2 wherein RIP engines 109 rip print job 103 into a printer ready format for RIPped output file 123 for output to print device 126 [paragraphs 0021 – 0022 and 0025 – 0026]*),

advising supervisor and collector that chunk processing completed (*see Figs. 1, 2, 16 and 17 wherein RIP engine(s) 109 report RIPping is complete to RIP manager 106 [i.e. supervisor] in step 756 [paragraph 0086] and wherein RIP handler 179 [i.e.*

collector] determines whether all partitions associated with a given print job 103 have been RIPped in step 703 [paragraph 0082]),

sending rasterized chunks to memory (*see Figs. 1, 2 and 16 wherein RIP manager 106 uploads rasterized partition 119 and stores it in memory that is local to the RIP manager 106 in step 686 [paragraphs 0024 and 0079]*),

completing print job by splitter (*see Figs. 1, 2 and 16 wherein RIP handler 179 applies RIPped output file 123 to a print device 126 or a print queue for printing in step 716 to complete the partitioned print job [paragraph 0083]*), and

advising supervisor of end-of-job by collector (*see Figs. 1, 2 and 16 wherein RIP handler 179 indicates to RIP manager 106 that print job 103 has been re-aggregated by updating the partition specification in step 713 [paragraphs 0026 and 0083]*) but does not explicitly disclose the method further comprising the steps of <u>selectively storing the jobs in the virtual disk remote transfer system for intermediary storage for data transfer to selected processing nodes, said system providing a shared memory interface.</u>

Cyr discloses a method of operating a printing system for parallel processing (*see Figs. 2 – 4 wherein the printer architecture utilizes network resources to distribute printer controller and translator functions and thereby process several print jobs [e.g. pre-rasterized images, which are equivalent to printer ready format for printing the print job] in parallel and wherein translators 12, 14, 16 are equivalent to a plurality of processing nodes [Abstract, column 3, lines 24 – 35 and 59 – 67]*) comprising the steps of:

<u>selectively storing the jobs in the virtual disk remote transfer system for intermediary storage for data transfer to selected processing nodes</u> (*see Figs. 3, 4, 6 and 10 wherein user interface job controller 66 utilizes VMS [virtual machine system] operating system and sends commands to print controller 25 as well as control process 67 of translator subsystems 12, 14, 16 to ensure an orderly flow of information between the components [column 4, lines 25 – 32, column 6, lines 19 – 24 and column 10, lines 57 – 67], see Figs. 4, 6 and 8 and 10 wherein print jobs or pre-rasterized images can be processed and stored while awaiting printing in storage device 11 in state 94 [column 5, lines 1 – 23, column 6, lines 19 – 31, 49 – 55 and column 7, lines 56 – 63] and see Figs. 4, 6 and 7wherein if the source document is pre-rasterized image in storage device 11, the print controller enters state 73 to access the stored image for transmission to printer interface 18 of state 76 [column 7, lines 18 – 22] such that the VMS operating system of job controller 66 controls storing/printing of jobs and print-ready pages [column 6, lines 19 – 24 and column 10, lines 57 – 67]*),

assigning the job chunks to respective processing nodes for parallel processing the jobs into the printer-ready format (*see Figs. 2, 4 and 6 wherein printer controller 25 assigns one of translators 12, 14, 16 to printing of a print request [column 6, lines 31 – 38 and 42 – 45]*), and

printing the print job (*see Figs. 2 and 4, printer 20 [column 3, lines 29 – 24 and column 5, lines 11 – 13], see Fig. 5, printer 48 [column 5, line 66 – column 6, line 4], see Fig. 6, printer engine 29 and see Fig. 7, state 76 [column 7, lines 23 – 31]*) but does not

explicitly disclose the method further comprising said system providing a shared memory interface.

IBM Technical Disclosure discloses the method wherein the virtual transfer system (*see Title*) data comprises an intermediary storage for data transfer to selected processing nodes, said system implemented by providing a shared memory interface (*"Disclosed is a program to create a shared disk for a cluster of processors which have separate I/O subsystems but a global shared memory. The program uses the shared memory to pass requests and data for I/O between processors, making it look like there is one shared disk rather than several disks that can only be accessed by the local processor." and "Separate disclosures have shown... that a shared virtual disk can be created to support shared access by all processors to disks, both local and remote, in the cluster"*).

Cyr and Christiansen are combinable because they are from the same field of endeavor, being high-speed image processing systems capable of printing large print jobs using multiple RIP engines. At the time of the invention, it would have been obvious to one of ordinary skill in the art to include splitting the job into a plurality of job chunks for processing of the print job at multiple RIP engines. The suggestion/motivation for doing so would have been to save time by partitioning very large documents, as suggested by Christiansen (*see Abstract and paragraph 0001*). See MPEP 2143 G.

Cyr and the IBM Technical Disclosure are combinable because they are from the same field of endeavor, being virtual disk remote transfer systems. At the time of the

invention, it would have been obvious to one of ordinary skill in the art to include a shared memory interface within a virtual disk remote transfer system. The suggestion/motivation for doing so would have been to make it look like there is one shared disk rather than several disks that can only be accessed by the local user and to provide fast access to non-permanent disk storage, as suggested by IBM Technical Disclosure. See MPEP 2143 G.

9. ***Claim 2*** is rejected under 35 U.S.C. 103(a) as being unpatentable over **Cyr *et al.*** (U.S. Patent No. 5,819,014) and in view of **Christiansen *et al.*** (U.S. Patent Application Publication No. 2004/0114170) in view of **IBM Technical Disclosure** NN9308637 and further in view of **Yomogizawa** (Japanese Patent No. 09-050354).

Referring to **claim 2**, Cyr, Christiansen and the IBM Technical Disclosure disclose the method using the VDISK system with shared memory interface as discussed above in claim 1 but does not explicitly disclose the method further comprising the step of preventing selected chunks from being added to the virtual disk transfer system when the monitored available space falls below a predetermined threshold representative of the overflow.

Yomogizawa discloses the method further comprising the step of preventing selected chunks from being added to the virtual disk transfer system when the monitored available space falls below a predetermined threshold representative of the overflow (*see Abstract wherein first printer only calls back the print which has been virtually spooled when a necessary space area is secured [i.e. the monitored available*

space falls below a predetermined threshold representative of the overflow] such that the received print data is virtually spooled in a remote printer whenever that received print cannot be received in the spool of the first printer [i.e. preventing selected chunks from being added]).

Cyr, Christiansen and Yomogizawa are combinable because they are from the same field of endeavor, being job management system within a network of printers. At the time of the invention, it would have been obvious to one of ordinary skill in the art to include preventing print data from being added to memory whenever overflow occurs. The suggestion/motivation for doing so would have been to provide a spool device for the received print data so long as one of the remote printers in the network has space whenever there becomes limited space in one local printer's spooler, as suggested by Yomogizawa (*see Abstract*). See MPEP §2143 G.

10. ***Claim 4*** is rejected under 35 U.S.C. 103(a) as being unpatentable over **Cyr *et al.*** (U.S. Patent No. 5,819,014) and in view of **Christiansen *et al.*** (U.S. Patent Application Publication No. 2004/0114170) in view of **IBM Technical Disclosure** NN9308637 and further in view of **Dimperio *et al.*** (U.S. Patent No. 5,142,667).

Referring to **claim 4**, Cyr, Christiansen and the IBM Technical Disclosure disclose the method as discussed above in claim 1 but does not explicitly disclose the method further including paging out the print data from the disk transfer system in a most-recently used order, wherein a least recently-used chunk is read soonest.

Dimperio discloses the method including paging out the print data from the disk transfer system in a most-recently used order, wherein a least recently-used chunk is read soonest (*see Figs. 20 – 21and 26 wherein most recent printing is best for deleting some image file(s, which may be pages or segments [e.g. parts of pages], from memory in order to make room for the next image file to be brought into memory [column 13, lines 19 – 33, 52 – 68, column 16, lines 30 – 40, 53 – 58, column 17, lines 3 – 30, column 18, lines 65 – 68 and column 22, lines 7 – 35]*).

Cyr, IBM Technical Disclosure and Dimperio are combinable because they are from the same field of endeavor, being memory management systems. At the time of the invention, it would have been obvious to one of ordinary skill in the art to include printing in most recently used order when a job is printing and memory reaches capacity. The suggestion/motivation for doing so would have been to improve memory and/or disk performance by generating the least amount of swapped segments while reducing the amount of disk reading to a minimum, as suggested by Dimperio (*column 13, lines 52 – 68, column 17, lines 20 – 28 and column 22, lines 17 – 35*). See MPEP 2143 G.

11. ***Claims 7 and 8*** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Cyr *et al.*** (U.S. Patent No. 5,819,014) and in view of **Christiansen *et al.*** (U.S. Patent Application Publication No. 2004/0114170) in view of **IBM Technical Disclosure** NN9308637 and further in view of **Yamazaki** (U.S. Patent No. 6,785,727).

Referring to **claims 7 and 8**, Christiansen discloses the method further wherein that other predetermined factors (*see Fig. 9, acceptance criteria 299*) may be employed and that the predetermined factors are not limited to that which is expressly described herein (*paragraph 0042*). However, neither Christiansen nor Yomogizawa explicitly disclose the method further wherein the predetermined factors used to determine the size of the print job further includes the total amount of processing required to process the job or the amount of startup time needed to set up the job.

Yamazaki discloses the method wherein the predetermined factors used to determine the size of the print job further includes the total amount of processing required to process the job or the amount of startup time needed to set up the job (*see Figs. 19 and 21 wherein a user sets button 1907 to the maximum allotted processing time necessary [i.e. amount of startup time needed] for an incoming print job [column 11, lines 15 – 20] and wherein a user sets buttons 1909, 1910 to split the incoming job into smaller jobs respectively having a size within a time required for processing [column 11, lines 41 – 52 and column 12, lines 6 – 11]*).

Cyr, Christiansen and Yamazaki are combinable because they are from the same field of endeavor, being job management system within a network of printers. At the time of the invention, it would have been obvious to one of ordinary skill in the art to include dividing print jobs into smaller processing times along with such print job management systems in a network. The design incentives, which would have prompted variations obvious to one of ordinary skill in the art, include optimizing the sharing of limited printer resources within a network by allowing certain manageable sized print

jobs of non-reserved print jobs to be processed while preventing useless delay in processing reserved print jobs, as suggested by Yamazaki (*column 11, lines 48 – 52*). In view of the identified design incentives and the disclosed adaptability of Christiansen, one of ordinary skill could have implemented the claimed variation of the prior art, and this variation would have been predictable to one of ordinary skill in the art. See MPEP 2143 F and G.

12. ***Claims 7 and 9*** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Cyr *et al.*** (U.S. Patent No. 5,819,014) and in view of **Christiansen *et al.*** (U.S. Patent Application Publication No. 2004/0114170) in view of **IBM Technical Disclosure** NN9308637 and further in view of **Wood *et al.*** (U.S. Patent Application Publication No. 2004/0243934).

Referring to **claim 7**, Christiansen discloses the method further wherein that other predetermined factors (*see Fig. 9, acceptance criteria 299*) may be employed and that the predetermined factors are not limited to that which is expressly described herein (*paragraph 0042*). However, neither Cyr, Christiansen nor the IBM Technical Disclosure explicitly disclose the method further wherein the predetermined factors used to determine the size of <u>the print</u> job further includes the total amount of processing required to process the job.

Wood discloses the method wherein the predetermined factors used to determine the size of <u>the print</u> job further includes the total amount of processing required to process the job (*see Figs. 1 and 4 wherein compound segmenting rules may*

include a rule which specifies that the PDL stream be divided into multi-page segments [i.e. chunks] along with another rule which specifies that graphic objects and text objects will be split out as segments in step 12 [paragraphs 0025 – 0026] and wherein scheduling process 24 assigns segments to processors based on the complexity of the segments and the processor's speed [paragraph 0036]).

One of ordinary skill in the art could have combined the elements as claimed by known methods of Christiansen and Wood, and that in combination, each element merely performs the same function as it does separately. The suggestion/motivation for the combination of elements would have been to provide improved parallel processing of multiple segments of a PDL data stream, as suggested by Wood (*paragraphs 0003, 0008 – 0009, 0011 – 0012, 0014 and 0036*). The combination yields nothing more than predictable results to one of ordinary skill in the art. See MPEP §2143 A and G.

Referring to **claim 9**, Christiansen discloses the method further wherein that other predetermined factors (*see Fig. 9, acceptance criteria 299*) may be employed and that the predetermined factors are not limited to that which is expressly described herein (*paragraph 0042*). However, neither Cyr, Christiansen nor the IBM Technical Disclosure explicitly disclose the method further wherein the predetermined factors used to determine the size of <u>the print</u> job further includes number of pages containing non-text images contained in the print job.

Wood discloses the method wherein the predetermined factors used to determine the size of <u>the print</u> job further includes number of pages containing non-text images contained in the print job (*see Figs. 1 and 4 wherein compound segmenting*

rules may include a rule which specifies that the PDL stream be divided into multi-page segments [i.e. chunks] along with another rule which specifies that graphic objects and text objects will be split out as segments in step 12 [paragraphs 0025 – 0026] and wherein scheduling process 24 assigns data files associated with text segments to processors that are customized for interpreting text data while assigning data files associated with image segments to processors that are customized for interpreting image data [Abstract and paragraph 0034] and see Fig. 5C wherein data files associated with text segments are sent to processor 38_1 while data files associated with black and white images and color images [i.e. non-text images] are sent to processors 38_2 and 38_3, respectively [Abstract and paragraph 0035]).

One of ordinary skill in the art could have combined the elements as claimed by known methods of Christiansen and Wood, and that in combination, each element merely performs the same function as it does separately. The suggestion/motivation for the combination of elements would have been to provide improved parallel processing of multiple segments of a PDL data stream, as suggested by Wood (*paragraphs 0003, 0008 – 0009, 0011 – 0012, 0014 and 0035*). The combination yields nothing more than predictable results to one of ordinary skill in the art. See MPEP §2143 A and G.

13. ***Claim 10*** is rejected under 35 U.S.C. 103(a) as being unpatentable over **Cyr *et al.*** (U.S. Patent No. 5,819,014) and in view of **Christiansen *et al.*** (U.S. Patent Application Publication No. 2004/0114170) in view of **IBM Technical Disclosure** NN9308637 and further in view of **Barry *et al.*** (U.S. Patent No. 5,859,711).

Referring to **claim 10**, Christiansen discloses the method further wherein that other predetermined factors (*see Fig. 9, acceptance criteria 299*) may be employed and that the predetermined factors are not limited to that which is expressly described herein (*paragraph 0042*). However, neither Cyr, Christiansen nor the IBM Technical Disclosure explicitly disclose the method further wherein the predetermined factors used to determine the size of the print job further includes number of pages containing color in the print job.

Barry discloses the method wherein the predetermined factors used to determine the size of the print job further includes number of pages containing color in the print job (*see Fig. 5 wherein the virtual printing system parses and routes pages of print jobs to engines that are more adapted to the particular printing operation [i.e. color vs. monochrome] associated with that particular page [column 8, line 30 – column 9, line 11, column 10, lines 1 – 19 and column 13, line 65 – column 14, line 47]*).

Cyr, Christiansen and Barry are combinable because they are from the same field of endeavor, being job management system within a network of printers. At the time of the invention, it would have been obvious to one of ordinary skill in the art to include dividing print jobs based upon pages of color and pages of black and white along with such print job management systems in a network. The suggestion/motivation for doing so would have been to provide for faster and more efficient printing of color jobs versus monochrome jobs, as suggested by Barry (*column 10, lines 15 – 19*). See MPEP §2143 G.

14. ***Claim 12*** is rejected under 35 U.S.C. 103(a) as being unpatentable over **Cyr *et al.*** (U.S. Patent No. 5,819,014) and in view of **Christiansen *et al.*** (U.S. Patent Application Publication No. 2004/0114170) in view of **IBM Technical Disclosure** NN9308637 and further in view of **Christiansen** (U.S. Patent Application Publication 2004/0196470).

Referring to **claim 12**, Cyr, Christiansen '170 and the IBM Technical Disclosure disclose the method as discussed in claim 1 but does not explicitly disclose the method further wherein the parallel processing system records the previous predetermined selection factors and uses statistical analysis to determine the optimal size of chunks.

Christiansen '470 discloses the method wherein the parallel processing system records the previous predetermined selection factors and uses statistical analysis to determine the optimal size of chunks (*see Figs. 2 – 3 wherein RIP manager 110 partitions based statistical data 222 and statistical analysis and scheduling module 304 [Abstract, paragraphs 0025, 0030, 0032 and 0033], see Fig. 4 wherein user interface 234 displays statistics gathered by module 304 [paragraphs 0042 and 0044] and see Fig. 6, steps 602 – 608 [paragraphs 0063 – 0066]*).

Cyr, Christiansen '170, and Christiansen '470 are combinable because they are from the same field of endeavor, being management system within a network of printers. At the time of the invention, it would have been obvious to one of ordinary skill in the art to include using statistical analysis to determine optimal size of chunks. The suggestion/motivation for doing so would have been to more accurately predict future

workload in part based upon historical workload levels, as suggested by Christiansen '470 (*paragraphs 0025 and 0030*). See MPEP 2143 G.

15. ***Claims 18 and 19*** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Cyr *et al.*** (U.S. Patent No. 5,819,014) and in view of **Christiansen *et al.*** (U.S. Patent Application Publication No. 2004/0114170) in view of **IBM Technical Disclosure** NN9308637 and further in view of **Cohen *et al.*** (U.S. Patent No. 6,356,355).

Referring to **claims 18 and 19**, Christiansen discloses parallel processing implemented using one or more processors and one or more memories (*see Fig. 2, processor(s) 143, memory 146 [paragraph 0028]*). However, neither Cyr, Christiansen nor the IBM Technical Disclosure does not explicitly disclose the method further wherein the parallel processing is implemented either in symmetric multiprocessing wherein two or more processors can connect to a single shared main memory or in distributed multiprocessor.

Cohen discloses the method wherein the parallel processing is implemented in symmetric multiprocessing wherein two or more processors can connect to a single shared memory (*see Fig. 2 wherein data processing system 200 may be a symmetric multiprocessor system including multiple processors 202, 204 sharing local memory 206 [column 3, lines 11 – 24]*). Claim scope is not limited by claim language that suggests or makes optional but does not require steps to be performed, or by claim language that does not limit a claim to a particular structure. Specifically, the claim language "can" suggests optional functionality or utility. See MPEP §2111.04.

Furthermore, Cohen discloses the method wherein the parallel processing is implemented in distributed multiprocessor (*see Fig. 1, distributed data processing system 100 [column 2, lines 1 – 3 and 38 – 50]*).

One of ordinary skill in the art at the time of the invention could have substituted the symmetric multiprocessing system or the distributed multiprocessor taught by Cohen for the parallel processing system comprising a plurality of processors and memories taught by Christiansen, and the results of the substitution would have been predictable. See MPEP 2143 B.

Conclusion

16. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Barry *et al.* (U.S. Patent No. 6,825,943) discloses a print system for parallel conversion processing of a print job using a plurality of RIP engines by partitioning the print job into select portions according to defined boundaries (*see Abstract and Figs. 1a – 2b, 4, 5b, 6, 9 and 10*).

Bender *et al.* (U.S. Patent No. 5,791,790) discloses a printer improving response time before jobs are printer-ready by providing a "fast data path" for certain print jobs while also storing on a hard disk all incoming print job data (*see Abstract and Figs. 2, 4, 5 and 7A – 7B*).

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Myles D. Robinson whose telephone number is (571)272-5944. The examiner can normally be reached on M-F 8:30am-5:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Twyler L. Haskins can be reached on (571) 272-7406. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Myles D. Robinson/
Examiner, Art Unit 2625
4/23/10

/Twyler L. Haskins/
Supervisory Patent Examiner, Art Unit 2625